

1. Match each type of generative modeling approach with the model's input and output. ("Thing" could be text, image, sound, etc.)
 - Autoregressive
 - Latent variable
 - Diffusion
 - Given a noisy whole thing, model outputs a less-noisy whole thing
 - Given a generation "cut off" somewhere, model outputs the next thing
 - Given a vector of random numbers, model outputs a complete sample in one step
 2. Describe how ChatGPT, internally, works like a classifier. What is it classifying? What are the classes?
-

1. Match each type of generative modeling approach with the model's input and output. ("Thing" could be text, image, sound, etc.)
 - Autoregressive
 - Latent variable
 - Diffusion
 - Given a noisy whole thing, model outputs a less-noisy whole thing
 - Given a generation "cut off" somewhere, model outputs the next thing
 - Given a vector of random numbers, model outputs a complete sample in one step
2. Describe how ChatGPT, internally, works like a classifier. What is it classifying? What are the classes?

Consider a chatbot conversation where the user asks “*What programming language should I learn?*” and the model responds with “*I recommend Python.*”

1. Write the “document” (the sequence of tokens) that was fed to the model to generate the word “*Python*”. Mark the boundaries between tokens with vertical bars. Include any special tokens needed.

2. Give an example of what the top-5 tokens in the next-token distribution might look like **just before** “*Python*” is generated. Make up reasonable probabilities that sum to 1.

Token	Probability
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

3. What does it mean that the model gives a probability distribution over language names here, rather than asking you follow-up questions? What should a good advisor do instead?

Consider a chatbot conversation where the user asks “*What programming language should I learn?*” and the model responds with “*I recommend Python.*”

1. Write the “document” (the sequence of tokens) that was fed to the model to generate the word “*Python*”. Mark the boundaries between tokens with vertical bars. Include any special tokens needed.

2. Give an example of what the top-5 tokens in the next-token distribution might look like **just before** “*Python*” is generated. Make up reasonable probabilities that sum to 1.

Token	Probability
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

3. What does it mean that the model gives a probability distribution over language names here, rather than asking you follow-up questions? What should a good advisor do instead?